

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (currently amended) A method comprising[;]

~~in a network, encapsulating data requests generated by an application in a first system,~~
~~encapsulating comprising~~

receiving data requests generated by an application executing on a first system;

generating an Extensible Markup Language (XML) structure for each data request; [~~and~~
~~converting the XML structure to an XML request;~~] the XML structure [~~comprising~~]
including

a variable stream of data stored in a memory of the first system, the stream
including

an XML element for each request, the XML element including

data from a data set object;

transferring the ~~encapsulated data requests~~ XML structure to a second system;

receiving a response from the second system;

updating the data set object based on the response; and

~~executing the encapsulated data requests in the second system; and~~

~~processing in the first system responses generated by the encapsulated data requests in~~
~~the second system.~~

updating controls in a graphical user interface based on the data set object.

2. (Canceled).

3. (Canceled).

4. (Previously presented) The method of claim 1 in which the XML element is a class object whose data is stored to generate XML.

5. (Canceled).

6. (Currently amended) The method of claim ~~5~~ 1 in which the data set object includes table dictionaries, column names and data from record sets, and stored procedure parameters.

7. (Currently amended) The method of claim 1, wherein transferring the XML structure comprises using ~~in which transferring includes~~ a text transmission protocol.

8. (Currently amended) The method of claim 7, in which the text transmission protocol is Hypertext Transfer Protocol (HTTP).

9. (Currently amended) The method of claim 1, ~~in which executing comprises~~ further comprising:

~~de-encapsulating the encapsulated data requests by parsing the XML structure~~
into request statements; and
executing the request statements.

10. (Original) The method of claim 9 further comprising:
translating responses from the executed request statements into an XML format; and
sending the XML formatted responses to the first system.

11. (Currently amended) A ~~distributed application~~ method for implementing a distributed application protocol, the method comprising:

~~[converting application requests in a first system, converting comprising]~~
receiving an application request from an application in a first system;
~~generating a data structure for storing data and parameters related to an application that produced the application requests,~~
translating the application requests into a data structure, the data structure being a
standardized delimited data structure stored in a memory of the first system, and
transforming the ~~[standardized delimited]~~ data structure ~~in conjunction with the data structure~~ into a stream of text based data utilizing an Extensible Markup Language (XML) format;
transmitting the ~~[converted application requests]~~ stream of text to a second system over a network, transmitting causing the second system to execute an executable command;
~~[parsing the converted application requests in the second system into a request statements; and]~~
~~[executing the request statements in the second system.]~~
receiving a response in the first system; and
adjusting the graphical user interface based on the response.

12. (Canceled).

13. (Currently amended) The method of claim 11, further comprising ~~[in which the parsing comprises]:~~

causing the second system to parse the stream of text by breaking down the
~~[converted application requests]~~ stream of text to an executable command format utilizing data and parameters related to an application.

14. (Currently amended) The method of claim 13, further comprising causing the second system to evaluate the ~~[in which executing further comprises evaluating]~~ executable command[s] prior to execution in the second system.

15. (Currently amended) The method of claim 14, further comprising causing the second system to evaluate [in which executing further comprises evaluating] a result[s] generated by executing the executable commands.

16. (Currently amended) The method of claim 15 further comprising:
causing the second system to convert[ing] the a result[s] into a stream of text based data
in a standardized XML format; and
transmitting the ~~converted~~ result[s] over the network to the first system.

17. (Currently amended) A~~n application server~~ method for implementing a distributed application protocol, the method comprising:
generating a first data structure for storing data and parameters [~~related to~~] received from
an application residing in the server, the first data structure [~~comprising~~] including database
tables, procedure results from logic calls and status/error messages;
translating application requests from the application into a delimited second data
structure, stored in a memory, the second data structure [~~comprising~~] having an element for each
of the application requests, the application requests being generated in response to user actions in
a graphical user interface; and
generating a stream of text-based[;] data in an Extensible Markup Language (XML)
format from the second data structure.

18. (Canceled).

19. (Canceled).

20. (Previously presented) The method of claim 17 in which the element is a class object.

21. (Currently amended) A method comprising:
in a server, receiving a stream of text-based data in an Extensible Markup Language (XML) format;

parsing the stream into request statements; ~~and~~
intercepting the request statements prior to execution; and
applying additional logic based on a type or content of the request.

22. (Canceled)

23. (Original) The method of claim 21 in which executing further comprises applying additional logic to responses generated from executing the request statements.

24. (Original) The method of claim 21 further comprising:
converting responses generated from each of the executed request statements into an XML format.

25. (Canceled).

26. (Currently amended) A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

generate a first data structure for storing data and parameters related to an application residing in the server, the first data structure comprising database tables, procedure results from logic calls and status/error messages;

translate application requests from the application into a delimited second data structure stored in a memory, the second data structure comprising an element for each of the application requests, the application requests being generated in response to user actions in a graphical user interface; and

generate a stream of text-based data in an Extensible Markup Language (XML) format from the second data structure.

27. (Previously presented) A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

receive a stream of text-based data in an Extensible Markup Language (XML) format;
parse the stream into request statements; and

intercept the request statements prior to execution and apply additional logic based on a type or content of the request .

Claims 28-34 (Canceled).